REMARKS/ARGUMENTS

Claims 1-20 are pending in the application.

Claims 1, 9, and 15 stand rejected under 35 U.S.C. §102(b) as being anticipated by "Simultaneous Multithreading: A Platform for Next-Generation Processor" by Eggers et al., (hereinafter "Eggers").

Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers.

Claims 3-8, 10-14, and 16-20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers in view of what the Examiner refers to as Applicants' admitted prior art, (hereinafter "AAPA").

Claim Rejections Under 35 U.S.C. §102(b)

Claims 1, 9, and 15 stand rejected under 35 U.S.C. §102(b) as being anticipated by Eggers.

As to claim 1, the Examiner has asserted that Eggers teaches the first instruction fetch unit and the second instruction fetch unit coupled to a multi-thread scheduler, referring to page 14, left column of Eggers. Applicants respectfully disagree because Eggers employs only one instruction fetch unit.

As the Examiner has noticed, a SMT fetch unit of Eggers partitions instruction bandwidth among threads. The unit has eight program counters, one for each thread context. On each cycle, it selects two different threads and fetches eight instructions from each thread, and chooses a subset of these instructions for decoding (Eggers,

page 14, line 3 from the bottom of the left column to line 3 of the right column). Eggers' thread selection hardware gives highest priority to the threads with the fewest instructions in the decoding, renaming and queue pipeline stages (Eggers, page 14, right column, lines 24-27). Thus, Eggers uses only one fetch unit to select threads. Eggers fails to teach the recited first and second instruction fetch unit. It is improper for the Examiner to read both the first instruction fetch unit and the second instruction fetch unit of claim 1 on the single fetch unit of Eggers.

The Examiner has also asserted that Eggers teaches the multi-thread scheduler which is to determine the width of an execute unit, as recited in claim 1. Applicants respectfully disagree. Eggers talks about simultaneous multithreading, but fails to teach the multi-thread scheduler which is to determine the width of an execution unit. Eggers uses dynamic scheduling hardware in current out-of-order superscalars for simultaneous multithreaded scheduling (Eggers, page 13, the last full paragraph of the right column). Eggers also states:

> In each cycle, the processor fetches eight instructions from the instruction cache. After instruction decoding, the register renaming logic maps the architectural registers to the hardware renaming registers to remove false dependencies. Instructions are then fed to either the integer or floating-point dispatch queues. When their operands become available, instructions are issued from these queues to their corresponding functional units. To support out-of-order execution, the processor tracks instruction and operand dependencies so that it can determine which instructions it can issue and which must wait for previously issued instructions to finish.

(Eggers, page 13, the paragraph bridging the columns).

Thus, in Eggers, the instructions are issued when their operands become available. Eggers does not mention width of the execution unit at all.

Accordingly, Applicants respectfully resubmit that claims 1-8 are patentable.

As to claims 9 and 15, the Examiner has asserted that the step of determining whether the processor is wide enough to execute the first and second threads in parallel is inherent in Eggers, arguing that because Eggers' processor exploits both thread-level and instruction-level parallelism and schedules to execute multiple threads together, the processor width must have been taken into account to determine whether the processor is wide enough to schedule and execute those threads in parallel. Applicants respectfully disagree.

To establish inherency, the extrinsic evidence " 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" *In re Robertson*, 169 F.3d 743 (Fed. Cir. 1999).

Here, the purpose of Eggers is to reduce horizontal and vertical waste of processors. Eggers does not address the opposite problem: what if the processor finds more instructions than it can execute in a cycle. Even assuming that the opposite problem would be recognized by a skilled artisan, there are other solutions, for example, by limiting the instruction width of the issue hardware, or by increasing the width of the processor.

Thus, it is not necessary for Eggers to determine whether the multi-threading processor is wide enough to execute the first and second thread in parallel, and the Examiner's assertion that this feature is inherent in Eggers is not supported.

Accordingly, claims 9 and 15 are patentable over Eggers.

Claims 2-7 are patentable for this additional reason as well.

Claim Rejections Under 35 U.S.C. §103(a)

Claim 2 stands rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers. As stated above, claim 2 is patentable over Eggers, because several features of claim 2 are missing from Eggers.

Claims 3-8, 10-14, and 16-20 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Eggers in view of the AAPA. As stated above, Eggers fails to teach or suggest the invention of claim 1, claim 9 and claim 15. The AAPA does not supply the deficiencies. Consequently, dependent claims 3-8, 10-14 and 16-20 are patentable.

The Examiner has argued that Eggers does not teach away from using an in-order processor because the processor of Eggers is able to schedule multiple threads when each have low instruction-level parallelism. Applicants respectfully disagree.

An in-order machine does not include hardware to determine instruction dependency, and instructions are executed in the same order that a compiler or program places them (Specification, page 3, the second full paragraph). However, Eggers uses dynamic scheduling hardware in current out-of-order superscalars for simultaneous multithread scheduling (Eggers, page 13, the first and second full paragraphs of the right

column). Accordingly, Applicants respectfully resubmit that Eggers teaches away from using an in-order processor, and is not a proper §103 reference of claims 3-8, 11-14, and 17-20.

It is also impropriate for the Examiner to combine Eggers, which uses out-of-order superscalars, with AAPA, which talks about in-order processors.

Thus, claims 3-8, 11-14, and 17-20 are patentable for this additional reason as well.

Request for Allowance

It is believed that this Response places the application in condition for allowance, and early favorable consideration of this Response is earnestly solicited.

If, in the opinion of the Examiner, an interview would expedite the prosecution of this application, the Examiner is invited to call the undersigned attorney at the telephone number listed below.

The Office is hereby authorized to charge any fees, or credit any overpayments, to Deposit Account No. **11-0600**.

Respectfully submitted,
KENYON & KENYON

Dated: October 31, 2005

By: Lin Deng
Lin Deng
(Limited Recognition No. L0239)
Attorneys for Intel Corporation

KENYON & KENYON
333 West San Carlos St.
San Jose, CA 95110

Telephone: (408) 975-7500
Facsimile: (408) 975-7501